Vol.2010-DPS-144 No.8
2010/9/18

排他的論理和書き込みによる NaryRAID の高速化

中村祐司[†]　　上原稔[†]

今日，ブログなどの CGM の普及や，写真ないし動画投稿サイトの流行により，ネットワーク上の情報が増大している．これらの情報は多量のディスクを用いた大規模ストレージの運用によりデータを保存しているが，ストレージのディスクが増えると信頼性が低下するという問題が発生する．ストレージの信頼性を向上させる技術として RAID があるが既存の RAID では 1 または 2 台の故障までしか修復できず信頼性が不十分で，階層型 RAID を用いれば十分な信頼性を得られるが容量効率が低くなるという問題点がある．我々は非階層で 3 耐故障を実現する NaryRAID を開発した. NaryRAID は基本 RAID より耐故障性に優れ, 階層型 RAID より容量効率が優れている．本研究では NaryRAID のパフォーマンスを向上させる方法とその効果の評価について述べる．

Faster for NaryRAID using WriteXOR

YUJI NAKAMURA[†]　MINORU UEHARA[†]

Today, CGM(Consumer Generated Media) such as blog is spread. In addition, photo and video sharing sites become popular. In this way, the amount of information in network increases drastically. Such information is usually stored into a large-scale storage which consists of many disks. In such a storage system, the larger the number of disks is, the lesser the reliability of the system. RAID is usually used to improve the reliability of a storage. However, it is not enough because conventional RAID repair only either one or two faults. Therefore, we require more than 2 fault tolerant RAID. If hierarchical RAID is employed, more than 2 fault tolerance can be realized but the capacity efficiency is decreased. We have proposed NaryRAID which realizes 3-fault tolerant without hierarchical organization. The capacity efficiency of NaryRAID is larger than that of hierarchical RAID. In this paper, we describe how to improve the performance of AAA and evaluate its effectiveness.

1. はじめに

今日，ブログなどの CGM の普及，写真ないし動画投稿サイトの流行などにより，かってないほどの情報がネットワーク上に氾濫している．Google はこれらの情報をインデックス化しようと努力しており，ある程度成功しているように見えるが，まだまだ未分類の情報が実世界に放置されている．最近では，日々の生活を記録する人々が増えている．これはライフログと呼ばれる．ライフログで動画が使われるようになると情報の量はいっそう増大する．仮想世界は実世界の情報をすべて取り込もうとしている．

仮想世界が実世界を飲み込もうとするとき，それらの情報を保存するストレージが必要になる．今日では，大量生産される安価な PC を用いて大規模なストレージを実現可能となった．また，HDD 技術の進歩によりストレージ容量も増えている．その結果，Amazon S3, Nirvanix などのオンラインストレージサービスが誕生した．SOA の視点から見てもオンラインストレージサービスはその他の Web サービスの基盤となる．本論文では，S3 のようなオンラインストレージサービスを大規模ストレージサービスと名づける．

大規模ストレージでは多量のディスクを運用する必要がある．しかし，ストレージのディスク数が増加すると信頼性が減少するという問題が生じる．一般にストレージの信頼性を増すには RAID を用いる．RAID の中でも RAID6 は 2 つのパリティで 2 台までの故障に耐える．商業製品では RAID6 が主流になりつつある．しかし，大規模ストレージサービスでは 2 耐故障の RAID6 でも不十分である．Accusys はパリティを 3 つに増やし 3 耐故障とした RAID6TP を開発したが，普及はしていない．

従来, 3 台以上の故障に耐える技法には RAID2 と階層型 RAID の HiRAID[1](全 RAID による多階層 RAID システム)しかなかった. RAID2 はシステムが複雑になるため実用化がされていない. HiRAID では 2 階層の RAID5 である RAID55 が 3 耐故障であるが，HiRAID には問題点として容量効率の低さがある．我々は RAID6 を超える 3 耐故障 RAID である NaryRAID を開発した[2][3][6]．NaryRAID は基本 RAID より耐故障性に優れ，階層型 RAID より容量効率が優れている．しかし読み書き性能において基本 RAID，階層型 RAID に劣っている．

本研究では，排他的論理和書き込みを行い，NaryRAID を高速化する．またその効果の評価についても行う．排他的論理和書き込みによる NaryRAID ではパリティ書き込みのためのリードデータの通信をリモートサーバクライアント間で行わないことで

[†] 所属
東洋大学大学院情報システム専攻
Dept. of Open Information Systems Graduate School of Engineering Toyo University, Japan

ⓒ 2010 Information Processing Society of Japan

書き込みの高速化を行っている．

本論文の構成は以下の通りである．2 節では NaryRAID について述べる．3 節では VLSD について述べる． 4 節では排他的論理和書き込みによる NaryRAID を実装する方法について述べる．6 節では排他的論理和書き込みによる NaryRAID の評価を行う．最後に結論を述べる．

2. NaryRAID

NaryRAID[2][3][6]は我々が開発した 3 耐故障 RAID 方式である．既存 RAID クラスでは，エラー訂正符号に基づく RAID2 に 3 以上の耐故障性がある．しかし，エラー訂正符号の計算は複雑であること，およびビット単位の計算であるため性能を重視する RAID には適さない．おそらく後者が最大の原因と考える．以上のことから 3 耐故障 RAID もブロック単位のパリティ方式でなければならない．

RAID6 は RAID5 より複雑であるが，RAID2 よりは単純である．RAID6 には P+Q 方式と 2D-XOR 方式がある．2D-XOR 方式は RAID5 と同様にストライプ内で水平パリティ(PH)をとるほか，別のストライプに対して対角パリティ(PD)をとる．PD がボトルネックとなり限界性能が低い．P+Q 方式ではストライプ内で 2 種のパリティをとるためボトルネックがない．そのためほとんどの商用 RAID6 は P+Q 方式を採用している．しかし，P+Q 方式を 3 耐故障に拡張することは容易ではない．そこで，本論文では，ボトルネックが発生しないように同一ストライプ内で複数のグループを構成しパリティをとる．グループは，N 進数の各桁に対応する．よって，この手法を N 進数(N-ary number)に基づく RAID として NaryRAID と名付ける．NaryRAID は基数 N とその次数 n で特徴付けられる．そこで，NaryRAID(N,n)と表記する．例えば N=2,n=3 の場合の NaryRAID は NaryRAID(2,3)とあらわす．

RAID4 を用いて 2 進 RAID の基本アイデアを説明する．2 進 RAID は 2 進数に基づく．2 進数の桁数を n とすると，n=3 における 2 進 RAID の構成を図 1 に示す．例えば，パリティ p0 はディスク d*i* のディスク番号 *i* の 1 桁目が 0 であるディスクグループのパリティである．

disk	d0	d1	d2	d3	d4	d5	d6	d7	0	1
2^0	0	1	0	1	0	1	0	1	p0	p1
2^1	0	0	1	1	0	0	1	1	p2	p3
2^2	0	0	0	0	1	1	1	1	p4	p5

図 1　BinaryRAID(n=3)
Figure 1 BinaryRAID(n=3)

n<3 のとき 2 進 RAID は 3 耐故障ではない．その原因は，データディスクとそのすべてのパリティディスクが同時に故障する可能性があるためである．ゆえに，すべてのデータディスクが必ず 3 つ以上のパリティグループに属せばよい．n>=3 のとき，その条件を満たす．

N 進 RAID は 2 進 RAID を単純に N 進数に拡張したものである．N 進 RAID では基数 N と最大データディスク数 Nn を示すレベル n で特徴づけることができる．よって，N 進 RAID のクラスを NaryRAID(N,n)と示すこととする．2 進 RAID は NaryRAID(2,n)である．図 2 に NaryRAID(3,3)の構成を紙面の都合上省略して示す．

	0	1	2	3	4	5	6	…	25	26	0	1	2
0	0	1	2	0	1	2	0	…	1	2	0	1	2
1	0	0	0	1	1	1	2	…	2	2	3	4	5
2	0	0	0	0	0	0	0	…	2	2	6	7	8

図 2　NaryRAID(3,3)
Figure 1 NaryRAID(3,3)

NaryRAID(3,3)のデータディスク数は 27，パリティディスク数は 6 である．一般に NaryRAID(N,n)のデータディスク数は Nn, パリティディスク数は Nn である．よって，その容量効率は以下の式で表せる．

$$\frac{N^n}{N^n + Nn} \qquad (2)$$

ここでは，NaryRAID(N,n)の修復アルゴリズムを示す．はじめに，NaryRAID の dep(p), rep(d)はそれぞれ以下の通りである．

$dep(p_j) = \{d_i \mid (i \text{ div } N^{(j \text{ div } N)}) \bmod N = (j \bmod N)\}$

$rep(d_i) = \cup_{k=0..n}\{p_j \mid j = Nk + (i \text{ div } N^k) \bmod N\}$

次に，アルゴリズム REPAIR_NARY_RAID を示す．このアルゴリズムは REPAIR を整理したもので，修復可能でなくても停止する．

Algorithm REPAIR_NARY_RAID
function P(i) : repair p_i using $dep(p_i)$ if $dep(p_i)$ is not empty; do nothing otherwise.
function D(i,j) : repair d_i using p_j and $dep(p_j) - \{d_i\}$.

ⓒ 2010 Information Processing Society of Japan

function R(i) : p_i if i-th disk is parity disk; rep(d_i) if i-th disk is data disk.
function E(x) : true if x is repairable, false otherwise.

```
begin
  S is a set of fault disks
  while ∃(x∈S) E(x) do
    case x
    x is parity → P(x) where dep(x) is not empty
    x is data → D(x,i)
        where p_i in rep(d_x) – ∪_{s∈S-{x}}R(s)
      esac
    end
    if S is not empty then failed to repair
  end
```

3 耐故障である NaryRAID と RAID55 を容量効率の面で比較する．容量効率を比較する．結果を図 3 に示す．ここで，x 軸は n を示し，RAID55(2)，RAID55(3)はそれぞれ N=2，3 としたときの NaryRAID と等しいディスク数の RAID55 における容量効率である．なお，RAID55 の容量効率は構成により異なるため，容量効率を最大化する構成で比較した．

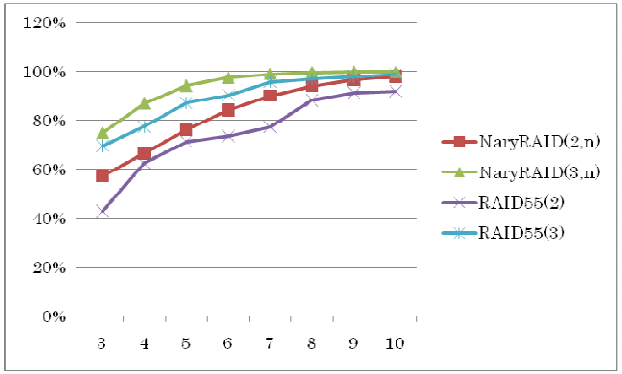

図 3　NaryRAID と RAID55 の容量効率
Figure 3. Capacity efficiency of NaryRAID and RAID55

図から明らかなように n が増えるほど容量効率は高まる．また，常に NaryRAID が RAID55 を上回る．よって，本方式は多数のディスクを用いる大容量ストレージに適する．

ここで，RAID55 の容量効率について考察する．RAID55 では上位層と下位層の数が等しいとき最大となる．すなわち全体の台数を N とするとそれぞれ√N となる．しかし，√N が整数にならないときは切り下げられ，それにより未使用のディスクが生じる．また，ストライピングに基づく RAID は同数のディスクで構成されなければならないため，そこでも端数が生じる．さらに，RAID55 はパリティのパリティも生成する．これらの損失によって RAID55 は容量効率を上げることができない．また，√N 以外の構成を採用すると容量効率はさらに下がる．
それに対して NaryRAID はデータディスク数が Nn でなくても構成できる．また，パリティのパリティを生成することはない．これらの理由から NaryRAID の方が HiRAID より容量効率がよい．

3. VLSD

本節では大規模ストレージ構築のための VLSD(Virtual Large Scale Disk)ツールキット[5]について述べる．VLSD は大規模ストレージ構築のためのツールキットであり，Java によるソフトウェア RAID 実装と NBD 実装を含む．VLSD は 100% pure Java であり，Java が動作するプラットフォームの上なら VLSD も動作する．そのため Windows や Linux が混在する環境に適している．

VLSD を用いると OS に制約されることなく NBD デバイスと RAID を自由に組み合わせることができる．最低限必要な NBD デバイスはファイルサーバの 1 つである．

Linux の nbd-server コマンドや Windows の nbdsrvr コマンドは単一ファイルを仮想ディスクとして公開する．そのため 4GB の制約がある FAT32 で動作させた場合，120GB/2GB=60 プロセスの NBD サーバを稼動させる必要がある．VLSD は複数のファイルを単一の JBOD にまとめて公開することができる．

ただし，VLSD の NBD サーバを用いた場合，ポート数の制約がある．ディスクを利用している最中はコネクションを維持するため NBD デバイスごとにポートを 1 つ消費する．ポート数はデバイス数より大きいため余裕があるが，その資源は無限ではない．数千台までは直接構成可能であるが，それを超える場合は間接的に，階層的に構成する必要がある．また，意図的に負荷を分散するために階層化することもある．この問題を解消するためにポート数に制限されない RMI を用いたディスクサーバも用意した．

ⓒ 2010 Information Processing Society of Japan

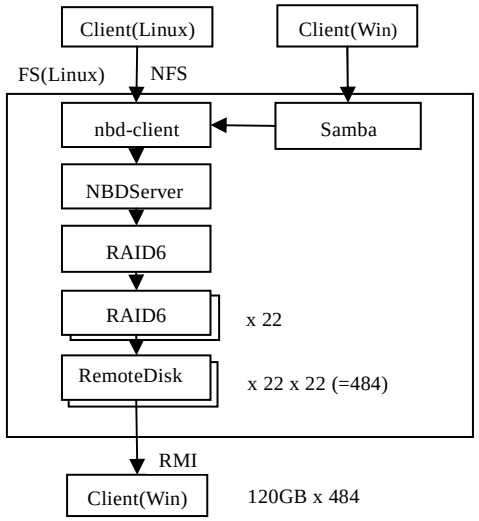

図 4　VLSD のシステム概要
Figure 4　The system overview of VLSD

図 4 に VLSD を用いて分散ストレージを構成した例を示す．クライアントは 500 台存在し，その OS は Linux または Windows である．それらはそれぞれ NFS，CIFS で 1 台のファイルサーバと通信する．クライアントは同時に NBD サーバでもある．各クライアントでは空き容量を束ねた 1 つの NBD サーバが稼動する（従来のシステムでは複数の NBD サーバを稼動させなければならない場合があった）．ファイルサーバは Samba の稼動する Linux マシンである．ファイルサーバでは，クライアントの分だけ NBDDisk（後述）を作成し，22 の NBDDisk から 1 つずつ合計 22 の RAID6 を作成し，最後に 22 の RAID6 から 1 つの RAID6 を作成する．この RAID0File を NBD サーバで公開し，自分自身の NBD デバイスで参照する．

VLSD ツールキットには以下のクラスが含まれる．

FileDisk

単一ファイルによる固定容量ディスク．論理的な容量と物理的な容量は正確に一致する．java.io.RandomAccessFile により実装される．

VariableDisk

単一ディスクにより容量可変ディスクを作成するラッパー．8KB を単位とする多分木で管理する．葉ノードには 8KB のデータが格納される．中間ノードには 1024 個の 64b(8B)ポインタが格納される．ノードは必要に応じて割り当てられる． 6 階層で 8EB-1 まで拡張できる．データ以外の管理情報が保存されるため物理的な容量は 0.1% 増加する．容量可変ディスクを実現するため，Disk インターフェースには容量を追加する API が定義されている．

NBDDisk/NBDServer

NBD デバイスのクライアント．NBDServer と NBD プロトコルで通信する．その他の NBD サーバ実装とも通信できる．

RemoteDisk/RemoteServer

遠隔デバイスのクライアント．RMI プロトコルで通信する．RemoteDisk に対応するサーバは DiskServer である．

SecureRemoteDisk/SecureRemoteServer

アクセスキーによる安全な遠隔デバイスのクライアント．RMI プロトコルで通信する．SecureRemoteDisk に対応するサーバは SecureDiskServer である．

JBOD

複数のディスクを直列に連結したディスク．冗長性がなく，容量増のために用いられる．各ディスクの容量は一様でなくてもよい．ストライピングを行わないため容量は単純に総和となる. 例えば, 100GB, 120GB, 160GB を連結すると 100+120+160=380GB になる．JBOD に対して連続的に逐次アクセスすると特定の部分ディスクに負荷が集中する．

RAID*n* (*n*=0,1,4,5,6)

各 RAID クラスの実装．RAID0 は HW RAID と異なり，JBOD と有意な差はない．RAID4, 5 は 1 耐故障である．RAID5 は HW RAID と異なり，RAID4 との有意な差はない．RAID6 は 2 耐故障である．P+Q 方式を採用している．

VotedRAID1

RAID1 に似ているが，多数決で任意故障をマスクする．書込み操作はすべてのディスクに複製される．読み取り操作はすべてのディスクに複製され，その結果を多数決する．多数決のため最低 3 台のディスクを必要とする．稼動ディスクが 2 台以下になると正しく多数決できなくなる．

これらのクラスは自由に組み合わせることができる．例えば，RAID6 を 2 段階で組み合わせると RAID66 を構築できる．

4. 排他的論理和書き込みによる NaryRAID の実装

VLSD に NaryRAID WriteXOR のクラスを追加し実装を行った．NaryRAID は RAID4

ⓒ 2010 Information Processing Society of Japan

のパリティをとるグループで構成するため, VLSD の RAID4 クラスを N 進数の各桁に対応するグループで構成するように改良し実装している.

以下に今回の実装に関係するプログラムの説明を行う.

RAID4

クラス RAID4 は RAID のサブクラスで RAID4 の実装（ブロック単位ストライピングとパリティ専用ディスクの作成）を行う．用意された要素ディスクで一番後ろのディスク番号のものをパリティディスクとして構築する.

NaryRAID

クラス NaryRAID は RAID のサブクラスで NaryRAID の実装である．用意された要素ディスクから指定した基数とレベルからデータディスクの台数とパリティディスクの台数を求め，ディスク番号 0 からデータディスクとして，データディスクの最後の番号に 1 足したディスク番号からパリティディスクとして NaryRAID を構築する.

NaryRAID WriteXOR

クラス NaryRAID WriteXOR は RAID のサブクラスで排他的論理和書き込みによる NaryRAID の実装である.

NaryRAID と NaryRAID WriteXOR には次の以下のメソッドを持つ.

repair

repair メソッドは故障した NaryRAID の要素ディスクの番号を格納した配列から修復可能な手順を導きだし，読み込みデータを計算する.

make_depends

make_depends メソッドはパリティディスクの番号からそのパリティディスクを構成するデータディスクの集合を求める 2 重配列 dep[j][]を作成する. dep[j][]はパリティディスク[j]を構成するデータで中にはデータディスク番号が格納される.

make_repairs

make_repairs メソッドはデータディスクの番号からそのデータディスクが持つパリティを求める 2 重配列 rep[no][k]を作成する．rep[no][k]はデータディスク[no]が作るパリティで中にはパリティ番号が収納される.

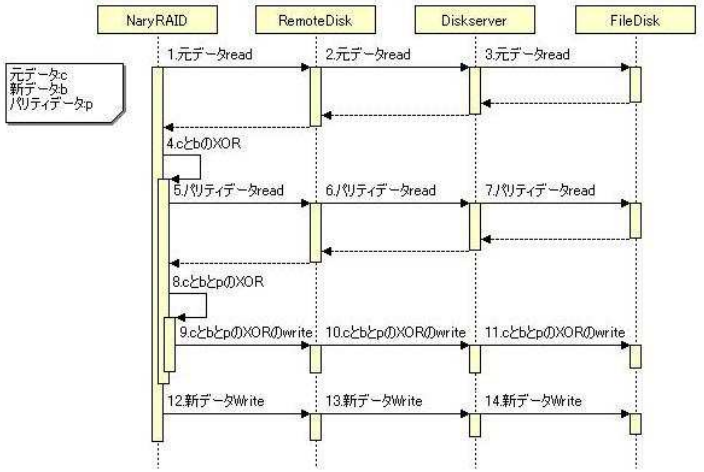

図 5　NaryRAID のディスク書き込み
Figure 5　Disk Writing of NaryRAID

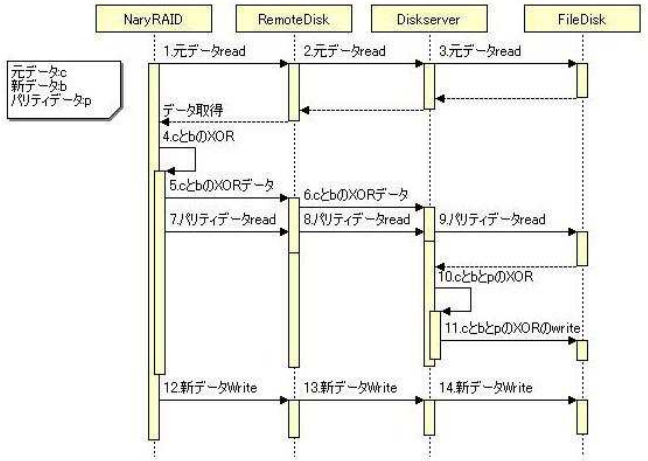

図 6　NaryRAID WriteXOR のディスク書き込み
Figure 6　Disk Writing of NaryRAID WriteXOR

ⓒ 2010 Information Processing Society of Japan

NaryRAID WriteXOR はパリティデータの更新を行うときに排他的書き込みを行いRemoteDisk と DiskServer 間でパリティデータの通信を行わないことで高速化を図っている. 図 5 と図 6 に従来の NaryRAID と NaryRAID WriteXOR の書き込み動作の例を示す．また 1 回の書き込みを行う際のデータの通信を NaryRAID(2,5)のときを例に表 1 に示す．NaryRAID は 1 つのディスクあたり次数と同じ数のパリティグループに所属しているので表のように NaryRAID WriteXOR では 5 つのパリティデータの読み込み分，大きく読み取り回数を減らすことができる．

表 1　NaryRAID と NaryRAID WriteXOR の読み書き回数
Table 1 The Read/Write number of times between NaryRAID and NaryRAID WriteXOR

	読み取り回数	書き込み回数
NaryRAID	6	6
NaryRAID WriteXOR	1	6

5. 評価

NaryRAID と排他的論理和書き込みによる NaryRAID の書き込み性能を評価する．NaryRAID と NaryRAID WriteXOR の書き込み時間の評価を行ったものを表 2 に示す．実験ではデータディスク 32 台パリティディスク 10 台で構成される NaryRAID(2,5)を 1 つのディスクサイズを 100MB として実験を行った．SmallWrite(SW)は同一ディスクに対する書き込みであり，LargeWrite(LW)はグループ全体におよぶ読み書きである．また，故障時の評価としてとして 3 台故障させたときの結果も SW3F と LW3F として示す．

実験の結果，無故障時と 3 台故障時ともに SW は NaryRAID と NaryRAID WriteXOR ではあまり差がないが，LW においては NaryRAID WriteXOR のほうが少ない時間で書き込みが行えることがわかる．

表 2　NaryRAID と NaryRAID WriteXOR の性能比較
Table 2. The comparison of performance between NaryRAID and NaryRAID WriteXOR

	SW(s)	LW(s)	SW3F(s)	LW3F(s)
NaryRAID(2,5)	47.911	103.046	45.516	96.889
NaryRAIDWriteXOR(2,5)	47.728	99.137	45.237	93.544

6. まとめ

本論文では排他的論理和書き込みによる NaryRAID の実装し評価を行った．評価では NaryRAID と NaryRAID WriteXOR を無故障時とディスク 3 台故障時の書き込み時間での性能評価を行った．その結果 1 つのディスクに対しての書き込みではほとんど差が出なかったがディスク全体の書き込みでは排他的書き込み方式の NaryRAID のほうが書き込み時間が少なく読み込み回数の減少の効果が得られ高速化が実現できている.

今後の課題としては，別の方法での NaryRAID の高速化を行い基本 RAID に劣っている読み書き性能の更なる改善を行う.その方法としては並列処理などが考えられる．また，文献[4]では複数パリティを持った NaryRAID について考察しているが，その実装を行う．実際に多くのマシンを用いて，運用評価を行う．

参考文献

[1] Sung Hoon Baek, Bong Wan Kim, Eui Joung Joung and Chong Won Park: “Reliability and Performance of Hierarchical RAID with Multiple Controllers,” In Proc. of 20th annual ACM symposium on principles of distributed computing, pp.246-254, (2001)
[2] 上原　稔 “2 進 RAID:もう一つの 3 耐故障 RAID”，信学技報 FIIS-08-228, (2008.3.7)
[3] Katsuyoshi Matsumoto, Minoru Uehara: "N-nary RAID: 3-resilient RAID based on an N-nary number", In Proceedings of 23rd International Conference on Advanced Information Networking and Applications(AINA2009), pp.249-255, (2008.5.26)
[4] Minoru Uehara: "Combining N-ary RAID to RAID MP", ITIS2009, (TBA)
[5] E, Chai, M, Uehara, H, Mori, N, Sato: “Virtual Large-Scale Disk System for PC-Room”, LNCS 4658, Network-Based Information Systems, pp.476-485, (2007.9.3-4)
[6] Yuji Nakamura, Minoru Uehara: "An Implementation of NaryRAID", In Proc. of 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA2010), pp.134-141, (Perth, Australia, 2010.4.20-23)

ⓒ 2010 Information Processing Society of Japan